

# SOFT INPUT SOFT OUTPUT DECODING METHOD AND SOFT INPUT SOFT OUTPUT DECODER

Patent number: JP2002344332

Publication date: 2002-11-29

Inventor: FUJITA HACHIRO; MIYATA YOSHIKUNI; NAKAMURA TAKAHIKO; YOSHIDA HIDEO

Applicant: MITSUBISHI ELECTRIC CORP

Classification:


- international: H03M13/45; G06F11/10; H03M13/29

- european:

Application number: JP20010145201 20010515

Priority number(s):

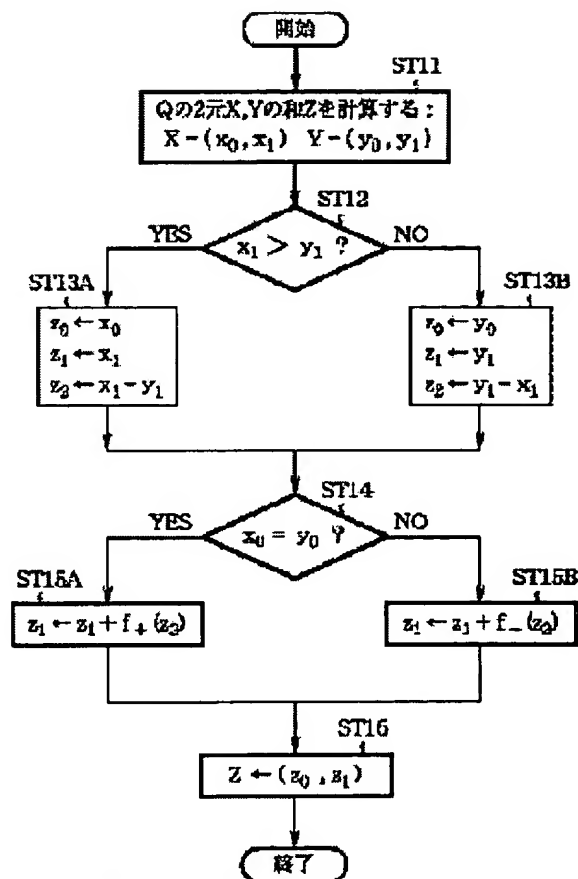
Also published as:

 JP2002344332 (A)

## Abstract of JP2002344332

**PROBLEM TO BE SOLVED:** To solve a problem that since the maximum number of states of an (N, K) two-element linear block code is generally  $2^{N-K}$  when a Trellis diagram of dual code is used, the number of states of Trellis diagram becomes tremendous when the number of check bits (N-K) increases and the complexity of recurrence formulas  $\alpha$  and  $\beta$  becomes tremendous, and a problem that the memory capacity for storing the recurrence formulas  $\alpha$  and  $\beta$  also becomes tremendous.

**SOLUTION:** The soft input soft output decoding method comprises a step to generate a first data sequence by subjecting hard decision data generated in a generating step to Hadamard transform and subjecting its reliability information to Hadamard transform to generate a second data sequence.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号  
特開2002-344332  
(P2002-344332A)

(43)公開日 平成14年11月29日(2002. 11. 29)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード <sup>*</sup> (参考)
H 0 3 M 13/45		H 0 3 M 13/45	5 B 0 0 1
G 0 6 F 11/10	3 3 0	G 0 6 F 11/10	3 3 0 S 5 J 0 6 5
H 0 3 M 13/29		H 0 3 M 13/29	

審査請求 未請求 請求項の数4 O L (全 18 頁)

(21)出願番号 特願2001-145201(P2001-145201)

(22)出願日 平成13年5月15日(2001. 5. 15)

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 藤田 八郎

東京都千代田区丸の内二丁目2番3号 三  
菱電機株式会社内

(72)発明者 宮田 好邦

東京都千代田区丸の内二丁目2番3号 三  
菱電機株式会社内

(74)代理人 100066474

弁理士 田澤 博昭 (外1名)

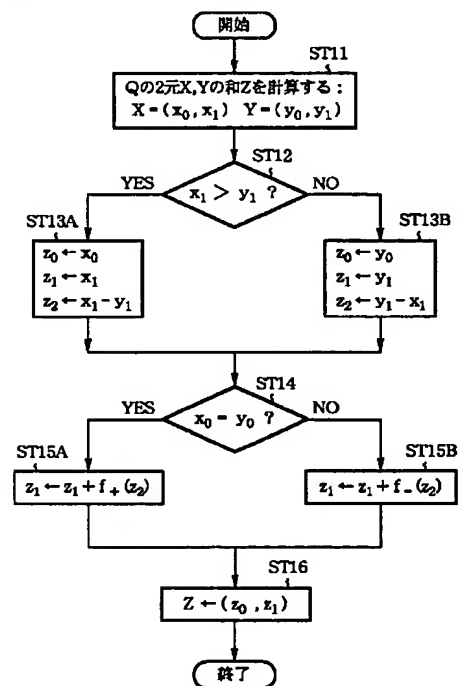
最終頁に続く

(54)【発明の名称】 軟入力軟出力復号方法及び軟入力軟出力復号装置

(57)【要約】

【課題】 双対符号のトレリス線図を用いて行われるが、一般に $(N, K)$  2元線形ブロック符号の状態数が最大で $2^{N-K}$ 個あるため、チェックビット数 $(N-K)$ が多いとトレリス線図の状態数が膨大となり、再帰式 $\alpha$ および $\beta$ の計算量が膨大になる課題があった。また、再帰式 $\alpha$ および $\beta$ を記憶するためのメモリ容量も膨大になる課題もあった。

【解決手段】 生成ステップで生成された硬判定データをアダマール変換して第1のデータ系列を生成し、その信頼度情報をアダマール変換して第2のデータ系列を生成するアダマール変換ステップを設ける。



【特許請求の範囲】

【請求項１】 通信路値と事前値から軟入力値を計算する軟入力値計算ステップと、上記軟入力値計算ステップで計算された軟入力値から硬判定データと信頼度情報を生成する生成ステップと、上記生成ステップで生成された硬判定データをアダマール変換して第１のデータ系列を生成し、その信頼度情報をアダマール変換して第２のデータ系列を生成するアダマール変換ステップと、上記アダマール変換ステップで生成された第１及び第２のデータ系列から外部値を計算する外部値計算ステップと、上記軟入力値計算ステップで計算された軟入力値と上記外部値計算ステップで計算された外部値を加算して事後値を計算する事後値計算ステップと、上記事後値計算ステップで計算された事後値から情報系列を推定する情報系列推定ステップとを備えた軟入力軟出力復号方法。

【請求項２】 事後値計算ステップは、２元線形ブロック符号Ｃ１、Ｃ２から構成された積符号を復号する場合、Ｃ１符号の事後値を計算すると、その事後値をＣ２符号の事前値として軟入力値計算ステップに与え、Ｃ２符号の事後値を計算すると、その事後値をＣ１符号の事前値として軟入力値計算ステップに与えることを特徴とする請求項１記載の軟入力軟出力復号方法。

【請求項３】 通信路値と事前値から軟入力値を計算する軟入力値計算回路と、上記軟入力値計算回路により計算された軟入力値から硬判定データと信頼度情報を生成する軟入力値変換回路と、上記軟入力値変換回路により生成された硬判定データをアダマール変換して第１のデータ系列を生成し、その信頼度情報をアダマール変換して第２のデータ系列を生成するアダマール変換回路と、上記アダマール変換回路により生成された第１及び第２のデータ系列から外部値を計算する外部値計算回路と、上記軟入力値計算回路により生成された軟入力値と上記外部値計算回路により計算された外部値を加算して事後値を計算する事後値計算回路と、上記事後値計算回路により計算された事後値から情報系列を推定する情報系列推定回路とを備えた軟入力軟出力復号装置。

【請求項４】 事後値計算回路は、２元線形ブロック符号Ｃ１、Ｃ２から構成された積符号を復号する場合、Ｃ１符号の事後値を計算すると、その事後値をＣ２符号の事前値として軟入力値計算回路に与え、Ｃ２符号の事後値を計算すると、その事後値をＣ１符号の事前値として軟入力値計算回路に与えることを特徴とする請求項３記載の軟入力軟出力復号装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は、２元線形ブロック符号を軟入力軟出力復号する軟入力軟出力復号方法及び軟入力軟出力復号装置に関するものである。

【０００２】

【従来の技術】積符号は訂正能力の比較的小さな符号から強力な誤り訂正符号を構成する手法として広く応用されている。図６は積符号の一般的な構成を示す説明図である。左上のブロック１は情報ビットのブロック（総数 $K_1 \cdot K_2$ ビット）であり、それ以外のブロック２、ブロック３およびブロック４はチェックビットのブロックである。各情報ビットは水平および垂直方向に２次的に符号化される。

【０００３】図６の垂直方向の符号は符号長 $N_1$ 、情報長 $K_1$ 、最小距離 $d_1$ の $(N_1, K_1, d_1)$ ２元線形符号（以降、Ｃ１符号と称する）であり、水平方向の符号は符号長 $N_2$ 、情報長 $K_2$ 、最小距離 $d_2$ の $(N_2, K_2, d_2)$ ２元線形符号（以降、Ｃ２符号と称する）である。このように、積符号は２次元符号化系列であり、全体で符号長 $N_1 \cdot N_2$ 、情報長 $K_1 \cdot K_2$ 、最小距離 $d_1 \cdot d_2$ の $(N_1 \cdot N_2, K_1 \cdot K_2, d_1 \cdot d_2)$ ２元線形符号となる（以降、符号Ｃと称する）。

【０００４】次に上述した積符号Ｃの符号化方法について説明する。情報データである $K_1 \cdot K_2$ ビット $\{d_{ij} \mid i=1, 2, \dots, K_1, j=1, 2, \dots, K_2\}$ （ $d_{ij}=0, 1$ ）は、式（１）に示すように、縦 $K_1$ 、横 $K_2$ のブロックＢ１に配列される。

【数１】

$$B1 = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,K_2-1} & d_{1,K_2} \\ d_{2,1} & d_{2,2} & \dots & d_{2,K_2-1} & d_{2,K_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{K_1-1,1} & d_{K_1-1,2} & \dots & d_{K_1-1,K_2-1} & d_{K_1-1,K_2} \\ d_{K_1,1} & d_{K_1,2} & \dots & d_{K_1,K_2-1} & d_{K_1,K_2} \end{bmatrix} \quad (1)$$

【０００５】まず、第１列から第 $K_2$ 列の各列ごとにＣ１符号化される。第 $j$ 列にはＣ１符号のパリティチェックビット $(N_1 - K_1)$ ビット $(r_{1,1,j}, r_{1,2,j}, \dots, r_{1,N_1-K_1,j})$ （ $j=1, \dots, K_2$ ）が付加される。このＣ１符号化により情報ビットの

ブロックＢ１の下に縦 $N_1 - K_1$ 、横 $K_2$ のチェックビットのブロックＢ２が配置されて全体で縦 $N_1$ 、横 $K_2$ のブロックＢ１２が生成される（式（２）を参照）。

【数２】

$$B12 = \begin{bmatrix} B1 \\ B2 \end{bmatrix}$$

(2)

$$B2 = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N2-K2-1} & r_{1,N2-K2} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N2-K2-1} & r_{2,N2-K2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{N1-K1-1,1} & r_{N1-K1-1,2} & \cdots & r_{N1-K1-1,N2-K2-1} & r_{N1-K1-1,N2-K2} \\ r_{N1-K1,1} & r_{N1-K1,2} & \cdots & r_{N1-K1,N2-K2-1} & r_{N1-K1,N2-K2} \end{bmatrix}$$

【0006】次にブロックB12の第1行から第N1行まで各行ごとにC2符号化される。第j行にはC2符号のパリティチェックビット(N2-K2ビット)( $r_{2,j,1}, r_{2,j,2}, \dots, r_{2,j,N2-K2}$ )( $j=1, \dots, N1$ )が付加される。これによりブロックB1に対して縦K1、横N2-K2のチェックビットのブ

ロックB3、また、ブロックB2に対して縦N1-K1、横N2-K2のチェックビットのブロックB4が生成されてブロックB12(B1+B2)の右に配置される(式(3)を参照)。以上の処理により積符号Cの符号化が完了する。

【数3】

$$B1234 = \begin{bmatrix} B1 & B3 \\ B2 & B4 \end{bmatrix}$$

$$B3 = \begin{bmatrix} r_{2,1} & r_{2,2} & \cdots & r_{2,N2-K2-1} & r_{2,N2-K2} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N2-K2-1} & r_{2,N2-K2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{2,N1-K1-1,1} & r_{2,N1-K1-1,2} & \cdots & r_{2,N1-K1-1,N2-K2-1} & r_{2,N1-K1-1,N2-K2} \\ r_{2,N1-K1,1} & r_{2,N1-K1,2} & \cdots & r_{2,N1-K1,N2-K2-1} & r_{2,N1-K1,N2-K2} \end{bmatrix}$$

(3)

$$B4 = \begin{bmatrix} r_{3,1} & r_{3,2} & \cdots & r_{3,N2-K2-1} & r_{3,N2-K2} \\ r_{3,1} & r_{3,2} & \cdots & r_{3,N2-K2-1} & r_{3,N2-K2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{3,N1-K1-1,1} & r_{3,N1-K1-1,2} & \cdots & r_{3,N1-K1-1,N2-K2-1} & r_{3,N1-K1-1,N2-K2} \\ r_{3,N1-K1,1} & r_{3,N1-K1,2} & \cdots & r_{3,N1-K1,N2-K2-1} & r_{3,N1-K1,N2-K2} \end{bmatrix}$$

【0007】積符号では、上述したように各情報ビットがC1符号およびC2符号により2重に符号化されているので強力な誤り訂正が可能である。積符号の復号方法には様々なものが知られている。まず、積符号の最小距離 $d = d_1 \cdot d_2$ の半分、即ち、 $t = (d-1)/2$ 個までの誤りを訂正する方法としてレディ・ロビンソン復号法が知られている。これは積符号の硬判定限界距離復号に相当する。詳細に関しては今井秀樹著『符号理論』(電子情報通信学会)に開示されている。

【0008】また、C1符号(垂直方向)の復号→C2符号(水平方向)の復号→C1符号(垂直方向)の復号→…のようにC1符号とC2符号を交互に繰り返して復号する方法もある。この繰り返し復号法を用いると上記のt個以上の誤りも訂正できる場合がある。これら2つの復号法は受信信号を硬判定した場合の復号法としてよく利用されている。

【0009】一方、積符号の軟入力軟出力繰り返し復号法も知られている。これは積符号を構成する要素符号C

1(またはC2符号)を復調器から入力される軟判定を用いて軟出力復号し、計算された軟出力を次のC2符号(またはC1符号)の復号の軟判定として軟出力復号を行い、再び計算された軟出力を次のC1符号(またはC2符号)の復号の軟判定として軟出力復号を行う、という処理を繰り返すものである。要素符号(C1符号またはC2符号)の軟入力軟出力復号法としては、J. Hagenauer他著, "Iterative decoding of binary block and convolutional codes", IEEE. IT, vol. 42, pp. 429-445, 1996に記載されたMAP復号法が知られている。

【0010】以下、上記論文に記載された(N, K)2元線形符号CのMAP復号法について説明する。MAP(Maximum A Posteriori probability、最大事後確率)復号法は軟判定復号法の1つで、通信路から得られる通信路情報を利用して復号ビット誤り率を最小にする復号法である。MAP復

号法を説明するために (N, K) 2元線形符号Cを用いた一般的なデジタル通信システムのモデルについて説明する。

【0011】図7は一般的なデジタル通信システムを示す構成図であり、図において、1は情報データに冗長ビットを付加して符号語を生成する符号器、2は符号器1で生成された符号語を通信路3に適した信号に変換する変調器、3は通信路、4は通信路3を介して入力された受信信号を復調し、その復調データを復号器5に出力する復調器、5は復調器4から入力された復調データを復号して推定情報ビット系列を出力する復号器である。なお、符号器1と変調器2から送信機が構成され、復調器4と復号器5から受信機が構成される。

【0012】次に動作について説明する。まず、長さKの情報ビット系列 $D = (d_1, d_2, \dots, d_K)$  ( $d_j = 0, 1$ )が符号器1に入力される。符号器1において入力された情報ビット系列Dに冗長ビット ( $r_1, r_2, \dots, r_{N-K}$ ) ( $r_j = 0, 1$ )が付加されて、長さNの符号語 $W = (d_1, d_2, \dots, d_K, r_1, r_2, \dots, r_{N-K})$ が生成される。記述を簡単にするために符号語Wを ( $w_1, w_2, \dots, w_N$ ) ( $w_j$

$= 0, 1$ )と表す。生成された符号語Wは変調器2に入力されて通信路3に適した信号に変換される。ここでは符号語の各ビット $w_j$ は、 $w_j = 0$ のとき“+1”に、 $w_j = 1$ のとき“-1”に変換されて通信路3に送出されるものとする。また、送信系列を $X = (x_1, x_2, \dots, x_N)$  ( $x_j = -1, +1$ )と表す。

【0013】通信路3では送信信号に加法的雑音が重畳されるものと仮定する。送信信号が通信路3を介して受信機に入力され、復調器4において受信信号が整形されて復調データ $Y = (y_1, y_2, \dots, y_N)$ が生成される。ここで復調データYの各成分は $y_j = x_j + n_j$  ( $n_j$ は雑音成分)と表される。復調器4で生成された復調データは復号器5に入力され、復号器5において入力された復調データからMAP復号により推定情報ビット系列(復号結果)が出力される。

【0014】以下、復号器5のMAP復号について詳細に説明する。復号器5では復調データ系列Yから下記の式(4)に示す送信ビット $w_j$ の対数尤度比 $L(w_j)$ を計算する。

【数4】

$$L(w_j) = \log \frac{\Pr(w_j = 0|Y)}{\Pr(w_j = 1|Y)} \quad (4)$$

【0015】式(4)の対数尤度比 $L(w_j)$ は、系列Yを受信した条件下で送信ビット $w_j$ が“0”であるか“1”であるかを判定するものである。対数尤度比L

( $w_j$ )が正ならば、送信ビット $w_j$ は“0”、負ならば“1”と判定される(式(5)を参照)。

【数5】

$$w_j = \begin{cases} 0 & \text{if } LLR(w_j) \geq 0 \\ 1 & \text{if } LLR(w_j) < 0 \end{cases} \quad (5)$$

【0016】式(5)の対数尤度比 $LLR(w_j)$ は、下記の式(6)に示すように3つの項に分解される。

$$LLR(w_j) = L_c(y_j) + L_a(w_j) + L_b(w_j) \quad (6)$$

第1項の $L_c \cdot y_j$ は、復調データ $y_j$ に比例する成分で通信路値と呼ばれる( $L_c$ は通信路の特性から定まる定数である)。第2項の $L_a(w_j)$ は、送信ビット“ $w_j$ ”の事前値と呼ばれる量で下記の式(7)で定義され

る。通常、送信ビットが“0”である確率と、“1”である確率は等しいと考えらるので、この項は0とされるが、繰り返し復号において重要な役割を演じる。

【数7】

$$L_a(w_j) = \log \frac{\Pr(w_j = 0)}{\Pr(w_j = 1)} \quad (7)$$

【0017】最後の第3項の $L_b(w_j)$ は、送信ビット“ $w_j$ ”の外部値と呼ばれる量で下記の式(8)で計算

される。

【数8】

$$L_b(w_j) = \log \frac{\sum_{i=1, i \neq j}^N \prod_{i=1, i \neq j}^N \tanh(L(w_i, y_i))^h}{\sum_{i=1, i \neq j}^N (-1)^{w_i} \prod_{i=1, i \neq j}^N \tanh(L(w_i, y_i))^h} \quad (8)$$

ただし、式(8)の右辺の記号 $C'$ は、符号Cの双対符号(符号長=N、情報長=N-K)で、 $B = (b_1, b$

$2, \dots, b_N)$ は双対符号 $C'$ の元である。双対符号に関しては前掲『符号理論』に開示されている。また、

$L(w_j, y_j)$ は、下記の式(9)で定義される通信路値と事前値の和である。式(8)の外部値は、積符号の繰返し復号において次段の要素符号のMAP復号にお

$$L(w_j, y_j) = L_e y_j + L_a(w_j)$$

【0018】式(8)は双対符号 $C'$ のトレリス線図を用いて計算することができる。符号 $C$ が生成多項式 $x^3 + x + 1$ で与えられる符号長7、情報長4の(7, 4)ハミング符号を例にとり、その双対符号のトレリス線図について説明する。なお、詳細は次の文献、J. K. Wolf著, "Efficient maximum likelihood decoding of linear block code using a trellis", IEEE Trans. IT, vol. 24, pp. 76-80, 1978に開示されている。

【0019】符号 $C$ の双対符号 $C'$ は、生成多項式が $x^4 + x^3 + x^2 + 1$ で与えられる符号長7、情報長3の2元巡回符号である。符号 $C'$ のトレリス線図は符号 $C'$ の符号器を用いて構成される。図8は(7, 3)巡回符号の符号器の構成を示すブロック図である。図中のD0からD3は1ビットを記憶する記憶素子、2A、2B、2Cは排他的論理和、SW1は情報ビットを入力するための第1のスイッチ、SW2は帰還接続するための第2のスイッチ、SW3は出力ビットを選択するための第3のスイッチである。

【0020】次に情報ビット系列( $d_1, d_2, d_3$ ) = (1, 0, 1)を符号化する場合を例にとり符号器の動作について説明する。まず、記憶素子D0からD3に0がセットされ、スイッチSW1、SW2がそれぞれ閉じた状態に設定され、SW3が端子1に接続される(時点0)。

【0021】次の時点1では $d_1 = 1$ がSW1を介して入力され、排他的論理和2Aにおいて記憶素子D0に記憶されている内容0と $d_1 = 1$ が加算されて、その出力1が帰還される。記憶素子D0には記憶素子D1に格納されている内容0と帰還された1が排他的論理和2Bにおいて加算されて、その出力1が格納される。また、記憶素子D1には記憶素子D2に格納されている内容0と帰還された1が排他的論理和2Cにおいて加算されて、その出力1が格納される。一方、記憶素子D2には記憶素子D3に格納されている内容0がそのままシフト入力され、記憶素子D3には帰還された1が入力される。また、第3のスイッチSW3では入力ビット $d_1 = 1$ が選択されて出力される。

【0022】時点2では入力ビット $d_2 = 0$ がスイッチSW1を介して入力され、排他的論理和2Aにおいて記憶素子D0に記憶されている内容1と $d_2 = 0$ が加算されて1が帰還される。記憶素子D0には記憶素子D1に格納されている内容1と帰還された1が排他的論理和2Bにおいて加算されて、その出力0が格納される。ま

ける事前値として再利用される。

【数9】

(9)

た、記憶素子D1には記憶素子D2に格納されている内容0と帰還された1が排他的論理和2Cにおいて加算されて、その出力1が格納される。一方、記憶素子D2には記憶素子D3に格納されている内容1がそのままシフト入力され、記憶素子D3には帰還された1が入力される。また、第3のスイッチSW3では入力ビット $d_2 = 0$ が選択されて出力される。

【0023】時点3では $d_3 = 1$ がSW1を介して入力され、排他的論理和2Aにおいて記憶素子D0に記憶されている内容0と $d_3 = 1$ が加算されて、その出力1が帰還される。記憶素子D0には記憶素子D1に格納されている内容1と帰還された1が排他的論理和2Bにおいて加算されて、その出力0が格納される。また、記憶素子D1には記憶素子D2に格納されている内容1と帰還された1が排他的論理和2Cにおいて加算されて、その出力0が格納される。一方、記憶素子D2には記憶素子D3に格納されている内容1がそのままシフト入力され、記憶素子D3には帰還された1が入力される。また、第3のスイッチSW3では入力ビット $d_3 = 1$ が選択されて出力される。時点3の処理が完了した段階で記憶素子に格納されている内容(D0, D1, D2, D3) = ( $r_1, r_2, r_3, r_4$ ) = (0, 0, 1, 1)は情報ビット系列( $d_1, d_2, d_3$ ) = (1, 0, 1)に対応するパリティチェックビットとなっている。

【0024】時点4ではSW1およびSW2が開いた状態に設定され、SW3は端子2に接続される。記憶素子D0には記憶素子D1の内容0がシフト入力され、記憶素子D1には記憶素子D2の内容1がシフト入力され、記憶素子D2には記憶素子D3の内容1がシフト入力される。また、記憶素子D3には0が格納される。SW3では記憶素子D0の内容、即ち、パリティチェックビット $r_1 = 0$ が選択されて出力される。以降、順に記憶素子の内容が出力側にシフトされてパリティビットが次々に出力される。時点7の処理が完了した段階で符号系列( $d_1, d_2, d_3, r_1, r_2, r_3, r_4$ ) = (1, 0, 1, 0, 0, 1, 1)がすべて出力され、記憶素子D0、D1、D2、D3の内容はすべて0にセットされる。

【0025】上の説明では情報ビット系列( $d_1, d_2, d_3$ ) = (1, 0, 1)の符号化について説明したが、他の情報ビット系列( $d_1, d_2, d_3$ )に対しても同様にパリティビット系列( $r_1, r_2, r_3, r_4$ )が生成される。以下では符号化系列( $d_1, d_2, d_3, r_1, r_2, r_3, r_4$ )を( $w$

$1, w_2, w_3, w_4, w_5, w_6, w_7)$ と表す。

【0026】次に符号C'のトレリス線図について説明する。上述した符号C'の符号器の各時点における記憶素子の内容(D0, D1, D2, D3)を符号器の状態と呼び、4ビットの数値D3D2D1D0で表す(D3をMSBとする)。上の例では時点1の状態はB、次の時点2の状態はEであり、符号器の状態は時点0から時点7まで0→B→E→C→6→3→1→0(16進数表現)のように変化している。

【0027】符号C'のトレリス線図はすべての符号系列の各時点における符号器の状態を時間軸にそって示したものである。図9は符号C'、即ち、(7, 3)巡回符号のトレリス線図を示す説明図である。次に図の構成法について説明する。横軸に時間軸、縦軸に0から15までの4ビットの数値を並べたグラフをTとし、時点0、状態0に口をプロットする。前述した符号化系列( $w_1, w_2, w_3, w_4, w_5, w_6, w_7$ ) = (1, 0, 1, 0, 0, 1, 1)はグラフT上で次のように表現される。

【0028】まず、時点1、状態Bの箇所に口をプロットし、時点0の状態0と時点1の状態Bを線(ブランチ

と呼ばれる)で結ぶ。また、結んだ線を出力ビット $w_1 = 1$ でラベル付けする。時点2以降に対しても同様の手続きを行う。即ち、時点kにおいて、その時の符号器の状態D3D2D1D0に口をプロットし、1時点前(k-1)の状態と現状態D3D2D1D0を線で結ぶ。また、結んだ線を時点kにおける出力ビット $w_k$ でラベル付けする。時点7における手続きが完了した段階で符号化系列( $w_1, w_2, w_3, w_4, w_5, w_6, w_7$ )の時点0から時点7までの符号器の状態遷移がグラフT上に表現される。上の符号系列(1, 0, 1, 0, 0, 1, 1)は図の太線の経路(パスと呼ばれる)で表現される。

【0029】同様にして、他の符号化系列に対して符号器の状態遷移をグラフTにプロットすれば、符号C'のトレリス線図が構成される。図からも分かるように符号化系列はすべて時点0において状態0から出発して時点7において状態0に終端する経路として表される。

【0030】符号C'のトレリス線図上で式(8)は次の式(10)のように変形される。

【数10】

$$L_t(w_t) = \log \frac{\sum_{e \in E_t^0} \alpha_{t-1}(i(e)) \beta_t(f(e)) + \sum_{e \in E_t^1} \alpha_{t-1}(i(e)) \beta_t(f(e))}{\sum_{e \in E_t^0} \alpha_{t-1}(i(e)) \beta_t(f(e)) - \sum_{e \in E_t^1} \alpha_{t-1}(i(e)) \beta_t(f(e))} \quad (10)$$

ただし、 $E_t^0$ は時点tにおいてラベル0のブランチの集合、 $E_t^1$ は時点tにおいてラベル1のブランチの集合を表し、 $i(e)$ はブランチeの出発側の状態、 $f(e)$ はブランチeの終着側の状態をそれぞれ表す。ブランチ

eのラベルを $l(e)$ で表し、各ブランチに対して次のようにブランチメトリックを定義する。

【0031】

【数11】

$$\Lambda(e) = \begin{cases} 1 & (l(e) = 0) \\ \tanh(L(w_t, y_t)) & (l(e) = 1) \end{cases} \quad (11)$$

式(10)の $\alpha_t(n)$ は式(11)のブランチメトリックを用いて時間の進む向きに再帰的に定義される量であ

る(式(12)を参照)。

【数12】

$$\alpha_t(n) = \sum_{e \in E_{t-1}(n)} \Lambda(e) \alpha_{t-1}(i(e)) \quad (12)$$

ただし、 $E_{t-1}(n)$ は状態nに終着するブランチの集合であり、式(12)の和はこの集合に含まれるすべてのブランチに対してとる。なお、時点0における $\alpha_0(n)$

の初期値は下記の式(13)で与えられる。

【数13】

$$\alpha_0(n) = \begin{cases} 1 & (n = 0) \\ 0 & (n \neq 0) \end{cases} \quad (13)$$

【0032】また、式(10)の $\beta_t(n)$ は、式(11)のブランチメトリックを用いて時間の戻る向きに再

帰的に定義される量である(式(14)を参照)。

【数14】

$$\beta_t(n) = \sum_{e \in E_t(n)} \Lambda(e) \beta_{t-1}(f(e)) \quad (14)$$

ただし、 $E_{t+1}(n)$ は状態nから出発するブランチの集合であり、式(14)の和はこの集合に含まれるすべ

てのブランチに対してとる。なお、時点7における $\beta_7(n)$ の初期値は下記の式(15)で与えられる。

【数15】

$$\beta_1(n) = \begin{cases} 1 & (n=0) \\ 0 & (n \neq 0) \end{cases} \quad (15)$$

【0033】次に式(12)で定義される $\alpha$ および式(14)で定義される $\beta$ の計算方法について図9を用いて具体的に説明する。 $\alpha$ および $\beta$ は同時に計算することもできるが、通常、 $\alpha$ を先に計算し、その後 $\beta$ の計算を行う場合が多い。まず、時点0において $\alpha_0(n)$  ( $n=0, 1, \dots, 15$ )を式(13)により初期化する。

【0034】時点1では図9のトレリス線図において時

一方、時点1の状態Bは時点0の状態0とブランチ1で結ばれているので、下記の式(17)のように計算され

【0035】時点2ではトレリス線図上で□のプロットされている状態0、5、B、Eに対して $\alpha_2(0)$ 、 $\alpha_2(5)$ 、 $\alpha_2(B)$ 、 $\alpha_2(E)$ を式(12)に従って同様に

$$\begin{aligned} \alpha_2(0) &= \Lambda(0)\alpha_1(0) \\ \alpha_2(5) &= \Lambda(1)\alpha_1(B) \\ \alpha_2(B) &= \Lambda(1)\alpha_1(0) \\ \alpha_2(E) &= \Lambda(0)\alpha_1(B) \end{aligned} \quad (18)$$

以降、時点3から時点7までトレリス線図上で□のプロットされている状態に対して同様に再帰式 $\alpha$ を計算する。なお、各時点で計算された再帰式 $\alpha$ はメモリに格納しておく。

【0036】 $\alpha$ の計算が完了すると次に $\beta$ の計算を開始する。 $\beta$ は時点の戻る向きに計算される。まず時点7に

$$\begin{aligned} \beta_7(0) &= \Lambda(0)\beta_6(0) \\ \beta_7(1) &= \Lambda(1)\beta_6(0) \end{aligned}$$

以降、時点5から時点0までトレリス線図上で□のプロットされている状態に対して同様に $\beta$ を計算する。また、各時点で計算された再帰式 $\beta$ はメモリに格納しておく。

【0037】 $\alpha$ および $\beta$ の計算が完了すると次に式(10)の外部値を時点1から順に計算する。例えば、時点

$$L_e(w_1) = \frac{\alpha_0(0)\beta_1(1) + \alpha_0(0)\beta_1(B)}{\alpha_0(0)\beta_1(1) + \alpha_0(0)\beta_1(B)}$$

【0038】このように双対符号Cのトレリス線図を用いて符号Cの外部値を計算する場合、再帰式 $\alpha$ および $\beta$ を計算しなければならないため計算量(特に乗算)が非常に多い。また、計算された $\alpha$ および $\beta$ を記憶するため、大きなメモリ容量も必要である。また、再帰式 $\alpha$ と $\beta$ 、外部値 $L_e$ の計算はすべて符号長に比例するため符号長が一般にNの場合、3Nの計算時間が必要である。

【0039】次に上記の軟入力軟出力繰り返し復号法を用いた積符号の軟入力軟出力繰り返し復号法について説明する。

点1において□のプロットされている状態0と状態Bに対して式(12)により $\alpha_1(0)$ と $\alpha_1(B)$ を計算する。時点1の状態0は時点0の状態0とブランチ0で結ばれているので、 $\alpha_1(0)$ は下記の式(16)のように計算される。

【数16】

る。 (16)

【数17】

計算すると、下記の式(18)のようになる。

【数18】

において $\beta_7(n)$  ( $n=0, 1, \dots, 15$ )を式(15)により初期化する。時点6では図9のトレリス線図の時点6において□のプロットされている状態0と状態1に対して式(14)に従って $\beta_6(0)$ と $\beta_6(1)$ を計算する(式(19)を参照)。

【数19】

(19)

1の符号ビット $w_1$ の外部値は $\alpha_0(0)$ 、 $\beta_6(1)$ および $\beta_1(B)$ をメモリから読み出して下記の式(20)のように計算される。時点2から時点7の符号ビットの外部値も同様に計算し符号系列の外部値を求める。

【数20】

(20)

説明を簡単にするためにC1符号とC2符号はともに符号長N、情報長Kの(N, K)2元線形符号とする。図10は積符号の軟入力軟出力繰り返し復号法を示すフローチャートであり、図において、ST1A、ST1Bはカウンタに初期値をセットするステップ、ST2A、ST2Bは通信路値と事前値から軟入力値を計算するステップ、ST3A、ST3Bは要素符号(C1符号およびC2符号)の軟入力軟出力復号を実行するステップ、ST4A、ST4BはステップST3A、ST3Bで計



算された軟出力値を事前値の配列に格納するステップ、ST5A、ST5Bはカウンターの値を判定するステップ、ST6A、ST6Bはカウンターをインクリメントするステップ、ST7は積符号の復号を繰り返すか否かを判定するステップである。

【0040】次に図10のフローチャートの動作について説明する。積符号の符号語Wを下記の式(21)のN×N行列で表す。

【数21】

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,N-1} & w_{1,N} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,N-1} & w_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{N-1,1} & w_{N-1,2} & \cdots & w_{N-1,N-1} & w_{N-1,N} \\ w_{N,1} & w_{N,2} & \cdots & w_{N,N-1} & w_{N,N} \end{bmatrix} \quad (21)$$

【0041】各列はC1符号の符号語であり、各行はC2符号の符号語である(行列の成分は0または1である)。上述した通信路モデルと同様にして、積符号Cの

符号語Wを送信し、受信機において復調データY(式(22)のN×N行列)が生成されたものとする。

【数22】

$$Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,N-1} & y_{1,N} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,N-1} & y_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{N-1,1} & y_{N-1,2} & \cdots & y_{N-1,N-1} & y_{N-1,N} \\ y_{N,1} & y_{N,2} & \cdots & y_{N,N-1} & y_{N,N} \end{bmatrix} \quad (22)$$

【0042】ここでYの各成分には通信路値の定数Lcが乗算されているものと仮定する。また、C1符号またはC2符号の軟入力軟出力復号において出力される外部

値を格納するためのN×N行列La(式(23))を用意し、初期値として0を格納しておく。

【数23】

$$La = \begin{bmatrix} La_{1,1} & La_{1,2} & \cdots & La_{1,N-1} & La_{1,N} \\ La_{2,1} & La_{2,2} & \cdots & La_{2,N-1} & La_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ La_{N-1,1} & La_{N-1,2} & \cdots & La_{N-1,N-1} & La_{N-1,N} \\ La_{N,1} & La_{N,2} & \cdots & La_{N,N-1} & La_{N,N} \end{bmatrix} \quad (23)$$

【0043】まず、ステップST1Aにおいてカウンタjに1をセットする。次にステップST2Aにおいて通信路値 $y_{1,j}$ と事前値 $La_{1,j} = La(w_{1,j})$ を加算して、式(24)のように、軟入力値 $R_{1,j} = R(w_{1,j})$ ( $j = 1, 2, \dots, N-1, N$ )を計算する。これは通

信路値の行列Yの第i行と事前値の行列Laの第i行を成分ごとに加算することに対応する(初回の事前値 $La(w_{1,j})$ はすべて0である)。

【数24】

$$R_{i,j} \leftarrow y_{i,j} + La_{i,j} (j = 1, 2, \dots, N-1, N) \quad (24)$$

【0044】ステップST3AではステップST2Aで計算された軟入力値 $R_{1,j}$ ( $j = 1, 2, \dots, N-1, N$ )から符号ビット $w_{1,j}$ ( $j = 1, 2, \dots, N-1, N$ )の外部値 $Le_{1,j}$ ( $j = 1, 2, \dots, N-1, N$ )を計算する。外部値の計算は上で述べた要素符号のMAP復号を適用し、C1符号の双対符号のトレリス線図を用いて行えばよい。ステップST4AにおいてステップST3Aで計算された外部値を配列Laの第1行に格納する。この場合、事前値として用いたデータに上書きすればよい。

で順に通信路値と事前値から軟入力値を計算し、計算された軟入力値から軟入力軟出力復号を実行して各符号ビットの外部値を計算し、事前値の配列Laに格納する。一方、カウンタの値がNであればステップST1Bに進む。この段階で配列Laの全成分の更新が完了している。

【0046】ステップST1Bではカウンタjに1をセットし、ステップST2Bに進む。ステップST2Bにおいて通信路値 $y_{1,1}$ と事前値 $La_{1,1}$ を加算し、軟入力値 $R_{1,1}$ を計算する( $i = 1, 2, \dots, N-1, N$ )。これは通信路値の行列Yの第1列と事前値の行列Laの第1列を成分ごとに加算することにより対応する。

【数25】

【0045】ステップST5Aにおいてカウンタの値がN未満であるかを判定し、N未満であればカウンタをインクリメントして(ステップST6A)、ステップST2A以降の処理を繰り返す。即ち、第2行から第N行ま

【0047】ステップST3BではステップST2Bで計算された軟入力値 $R_{i-1}$  ( $i=1, 2, \dots, N-1, N$ ) から符号ビット $w_{i-1}$  ( $i=1, 2, \dots, N-1, N$ ) の外部値 $L_{e_{i-1}}$  ( $i=1, 2, \dots, N-1, N$ ) を計算する。外部値の計算は上で述べた要素符号のMAP復号と同様にC2符号の双対符号のトレリス線図を用いて行う。ステップST4BにおいてステップST3Bで計算された外部値を配列Laの第1列に格納する。この場合、事前値として用いたデータの上に書きすべし。

【0048】ステップST5Bにおいてカウンタjの値がN未満であるかを判定し、N未満であればカウンタをインクリメントして(ステップST6B)、ステップST2B以降の処理を繰り返す。即ち、第2列から第N列まで同様に通信路値と事前値から軟入力値を計算し、計算された軟入力値から軟入力軟出力復号により外部値の計算を行い、事前値の行列Laに格納する。一方、カウンタ値がNであればステップST7に進む。この段階で配列Laの全成分が更新されている。

【0049】以上の処理により積符号全体の1回の繰り返し復号が完了する。ステップST7においてさらに繰り返し復号するか否かの判定を行う。繰り返し復号場合はステップST1Aに進み、繰り返ししない場合は処理を終了する。通常、誤りがすべて訂正されるか、または、所定の回数の繰り返し復号が完了した段階で復号処理を終了する。

【0050】従来の積符号の軟入力軟出力繰り返し復号法は以上のように構成されているので、各要素符号の軟入力軟出力復号に要する処理時間が膨大であるため積符号全体の復号を完了するまでに多大の時間を必要とする。

【0051】

【発明が解決しようとする課題】従来の軟入力軟出力復号方法は以上のように構成されているので、双対符号のトレリス線図を用いて行われるが、一般に $(N, K)$  2元線形ブロック符号の状態数が最大で $2^{N-K}$ 個あるため、チェックビット数 $(N-K)$ が多いとトレリス線図の状態数が膨大となり、再帰式 $\alpha$ および $\beta$ の計算量が膨大になる課題があった。また、再帰式 $\alpha$ および $\beta$ を記憶するためのメモリ容量も膨大になる課題もあった。さらに、従来の積符号の軟入力軟出力繰り返し復号法ではC1符号およびC2符号を繰り返して軟入力軟出力復号するために復号結果が求められるまでの処理時間が膨大になる課題があった。

【0052】この発明は上記のような課題を解決するためになされたもので、2元線形ブロック符号の軟入力軟出力復号を少ない計算量とメモリ容量で行うことができる軟入力軟出力復号方法及び軟入力軟出力復号装置を得ることを目的とする。また、この発明は、積符号の軟

力軟出力復号を少ない計算量とメモリ容量で行うことができる軟入力軟出力復号方法及び軟入力軟出力復号装置を得ることを目的とする。

【0053】

【課題を解決するための手段】この発明に係る軟入力軟出力復号方法は、生成ステップで生成された硬判定データをアダマール変換して第1のデータ系列を生成し、その信頼度情報をアダマール変換して第2のデータ系列を生成するアダマール変換ステップを設けたものである。

【0054】この発明に係る軟入力軟出力復号方法は、事後値計算ステップが2元線形ブロック符号C1、C2から構成された積符号を復号する場合、C1符号の事後値を計算すると、その事後値をC2符号の事前値として軟入力値計算ステップに与え、C2符号の事後値を計算すると、その事後値をC1符号の事前値として軟入力値計算ステップに与えるようにしたものである。

【0055】この発明に係る軟入力軟出力復号装置は、軟入力値変換回路により生成された硬判定データをアダマール変換して第1のデータ系列を生成し、その信頼度情報をアダマール変換して第2のデータ系列を生成するアダマール変換回路を設けたものである。

【0056】この発明に係る軟入力軟出力復号装置は、事後値計算回路が2元線形ブロック符号C1、C2から構成された積符号を復号する場合、C1符号の事後値を計算すると、その事後値をC2符号の事前値として軟入力値計算回路に与え、C2符号の事後値を計算すると、その事後値をC1符号の事前値として軟入力値計算回路に与えるようにしたものである。

【0057】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態1. 以下、この発明の実施の形態1による2元線形ブロック符号の軟入力軟出力復号方法について説明する。ここでは、符号長 $N=2^m$ 、情報長 $K=2^m-1-m$  ( $m$ は正の整数)の $(N, K)$  拡大ハミング符号を用いて説明する。また、以下の説明では、この拡大ハミング符号を符号Cと称する。符号Cの双対符号(C'とする)は符号長 $N=2^m$ 、情報長 $m+1$ の1次のリード・マラー符号である。符号長Nの1次のリード・マラー符号はN次のアダマール行列 $H_N$ から構成することができる。 $m=3$ の場合を例にとり具体的に説明する。

【0058】符号長 $N=2^m=8$ 、情報長 $K=m+1=4$ の1次のリード・マラー符号の符号語は、下記の式(26)の8次のアダマール行列 $H_8$ と、その反転行列 $-H_8$ において、行列要素“+1”をビット“0”に、“-1”をビット“1”に変換した行列の行に対応している(したがって全部で $2N=16$ 行)。

【数26】

$$H_8 = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{bmatrix}, -H_8 = \begin{bmatrix} -h_1 \\ -h_2 \\ -h_3 \\ -h_4 \\ -h_5 \\ -h_6 \\ -h_7 \\ -h_8 \end{bmatrix} = \begin{bmatrix} h_9 \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{13} \\ h_{14} \\ h_{15} \\ h_{16} \end{bmatrix}$$

(26)

なお、リード・マラー符号の詳細については前掲『符号理論』参照。

【0059】以下では符号長をNとして一般的に説明する。N次のアダマール行列 $H_N$ を構成する行ベクトルを上から $h_1, h_2, \dots, h_N$ と表し、反転行列 $-H_N$ の行ベクトルを上から $h_{N+1}, h_{N+2}, \dots, h_{2N}$ と表す(式(26)では $N=8$ )。また各行 $h_m$  ( $m=1, 2, \dots, 2N-1, 2N$ )に対応するリード・マラー符号の符号語を $B_m$  ( $m=1, 2, \dots, 2N-1, 2N$ )と表す。逆にいえば、符号語 $B_k = (b_1, b_2, \dots, b_{N-1}, b_N)$  ( $b_m=0, 1$ )の各ビット“ $b_k$ ”を実数“ $(-1)^{b_k}$ ”に変換した行ベクトルが $h_k$ である。このとき、特に双対符号 $C'$ の零元(0,

0, ..., 0, 0)はアダマール行列 $H_N$ の第1行 $h_1$ に対応している。

【0060】この実施の形態1では、従来技術の箇所で説明した図7の通信路モデルを仮定して説明する。送信機において拡大ハミング符号Cの符号語 $W = (w_1, w_2, \dots, w_{N-1}, w_N)$ が送信され、通信路3を介して受信機において復調データ $Y = (y_1, y_2, \dots, y_{N-1}, y_N)$ が生成されたものとする。復調データ $Y = (y_1, y_2, \dots, y_{N-1}, y_N)$ に対して実数系列 $u_m, v_m$  ( $m=1, 2, \dots, N-1, N$ )を次の式(27)、式(28)により定義する。ただし、 $\epsilon$ は十分小さい正数である。

【数27】

$$u_m = \begin{cases} 0 & (y_m \geq 0) \\ 1 & (y_m < 0) \end{cases} \quad (27)$$

$$v_m = \begin{cases} |\tanh(L, y_m)| & (|y_m| > \epsilon) \\ |\tanh(L, \epsilon)| & (|y_m| \leq \epsilon) \end{cases} \quad (28)$$

ただし、式(28)の $\tanh$ は次の式(29)で定義される双曲正接関数である。

【数28】

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (29)$$

【0061】双対符号 $C'$ の符号語 $B_k = (b_1, b_2, \dots, b_{N-1}, b_N)$  ( $b_m=0, 1$ )に対して、下記の式(30)および式(31)により $U(B_k)$

および $V(B_k)$ を定義する。

【数29】

$$U(B_k) = \frac{1}{2} \sum_{m=1}^N (2b_m - 1) \cdot u_m = -\frac{1}{2} \sum_{m=1}^N (-1)^{b_m} \cdot u_m \quad (30)$$

$$V(B_k) = \prod_{m=1}^N v_m^{b_m - 1/2} \quad (31)$$

【0062】このとき、下記の式(32)の関係に注意すると、式(8)に現れる積の項 $\prod \tanh()$ は式

(33)により近似される。

【数30】

$$U(B_k) - U(B_1) = \sum_{m=1}^N b_m u_m \quad (32)$$

$$\begin{aligned} \prod_{l=1, l \neq m}^N \tanh(L_\epsilon y_l)^{b_l} &\approx \prod_{l=1, l \neq m}^N (-1)^{b_l u_l} v_l^{b_l} \\ &= (-1)^{U(B_k) - U(B_1)} \left( (-1)^{u_m} v_m^{-1} \right)^{b_m} \prod_{l=1}^N v_l^{b_l - 1/2} \prod_{l=1}^N v_l^{1/2} \\ &= (-1)^{U(B_k) - U(B_1)} \left( (-1)^{u_m} v_m^{-1} \right)^{b_m} V(B_k) V(B_{N+1}) \\ &= \left( (-1)^{u_m} v_m^{-1} \right)^{b_m} (-1)^{-U(B_1)} V(B_{N+1}) Z(B_k) \end{aligned} \quad (33)$$

【0063】ただし、式(33)中の $Z(B_k)$ は、下記の式(34)で定義される双対符号 $C'$ の符号語 $B_k$ に

【数31】

$$Z(B_k) = (-1)^{U(B_k)} V(B_k) \quad (34)$$

式(33)の近似値を用いると、式(8)の外部値は、次の式(35)のように近似される。

【数32】

$$L_\epsilon(w_m) \approx \log \frac{\sum_{B_k \in \mathcal{B}_m=0} Z(B_k) + \sum_{B_k \in \mathcal{B}_m=1} Z(B_k) (-1)^{u_m} v_m^{-1}}{\sum_{B_k \in \mathcal{B}_m=0} Z(B_k) - \sum_{B_k \in \mathcal{B}_m=1} Z(B_k) (-1)^{u_m} v_m^{-1}} \quad (35)$$

【0064】また、下記の式(36)および式(37)により $S^{(0)}_m$ と $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ )を定義すると、式(35)は式(38)のように表

される。

【数33】

$$S_m^{(0)} = \sum_{B_k \in \mathcal{B}_m=0} Z(B_k) \quad (36)$$

$$S_m^{(1)} = \sum_{B_k \in \mathcal{B}_m=1} Z(B_k) (-1)^{u_m} v_m^{-1} \quad (37)$$

$$L_\epsilon(w_m) \approx \log \frac{S_m^{(0)} + S_m^{(1)}}{S_m^{(0)} - S_m^{(1)}} \quad (38)$$

【0065】式(38)のLLRを計算するには、式(36)、式(37)で定義される $S^{(0)}_m$ と $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ )を求める必要がある。一方、 $S^{(0)}_m$ と $S^{(1)}_m$ を計算するには、式(34)で定義される $Z(B)$  ( $B$ は双対符号の符号語)を求める必要がある。

【0066】まず、 $Z(B)$  ( $B$ は双対符号の符号語)の

計算方法について説明する。双対符号の符号語 $B$ に対応する $Z(B)$ は同じ符号語 $B$ の $U(B)$  (式(30))および $V(B)$  (式(31))から構成される。まず $U(B)$ の計算方法について説明する。式(27)で定義される実数系列 $u_m$  ( $m=1, 2, \dots, N-1, N$ )を並べて列ベクトル $U$ を定義する(式(39))。

【数34】

$$U = {}^T(u_1, u_2, \dots, u_N) \quad (39)$$

【0067】式(32)により、符号語 $B_k$  ( $k=1, 2, \dots, N$ )に対応する $U(B_k)$ は行ベクトル $h_k$ と列ベクトル $U$ の積、即ち、下記の式(40)の内積で計算

される。

【数35】

$$U(B_k) = -\frac{1}{2} h_k \cdot U \quad (40)$$

式(40)を各 $k$  ( $k=1, 2, \dots, N$ )についてまとめると、列ベクトル $U$ のアダマール変換 $(-1/2)H_N$

$U$ が得られる。よって、列ベクトル $U$ のアダマール変換 $H_N U$ を計算すれば、双対符号の符号語半分について $U$

( $B_k$ ) ( $k=1, 2, \dots, N$ ) が計算できる。一方、残りの符号語  $B_k$  ( $k=N+1, N+2, \dots, 2N$ ) に対応する  $U(B_k)$  は、次の式 (41) により上で計算した

$$U(B_k) = -\frac{1}{2}h_k \cdot U = \frac{1}{2}h_{k-N} \cdot U = -U(B_{k-N}) \quad (41)$$

【0068】次に式 (31) の  $V(B_k)$  の計算方法について説明する。 $V(B_k)$  の対数を取り、下記の式 (4

$U(B_{k-N})$  を  $-1$  倍すれば求められる。  
【数36】

2) のように  $V'(B_k)$  とおく。  
【数37】

$$V'(B_k) = \log V(B_k) = \frac{1}{2} \sum_{i=1}^N (2b_{ki} - 1) \log v_i = -\frac{1}{2} \sum_{i=1}^N (-1)^{b_{ki}} \log v_i \quad (42)$$

列ベクトル  $V'$  を下記の式 (43) により定義すると、 $V'(B_k)$  ( $k=1, 2, \dots, N$ ) は  $U(B_k)$  ( $k=1, 2, \dots, N$ ) と同様に、列ベクトル  $V'$  のアダマール変換のスカラー倍  $(-1/2)H_B V'$  から求められる。また、 $V'(B_k)$  ( $k=N+1, N+2, \dots, 2$

$N$ ) は  $V'(B_{N-k})$  を  $-1$  倍すれば求められることも  $U(B_k)$  の場合と同様である。なお、指数関数を利用すれば  $V'(B_k)$  から  $V(B_k) = \exp V'(B_k)$  を計算できる。

【数38】

$$V'^T (\log v_1, \log v_2, \dots, \log v_N) = (v'_1, v'_2, \dots, v'_N) \quad (43)$$

【0069】このように  $Z(B_k)$  に現れる  $U(B_k)$ 、 $V(B_k)$  は、ともに  $N$  次のアダマール変換  $H_N$  を利用して計算することができる。 $N$  次のアダマール変換  $H_N U$  および  $H_N V'$  は一般的な行列の演算法則により計算することもできるが、アダマール行列の特徴を巧みに利用した効率的な計算方法 (グリーン・マシーン) が知られている。アダマール変換の効率的な計算方法については宮川洋他著『符号理論』、F. J. MacWilliams 他著『The Theory of Error-Correcting Codes』(North-Holland) を参照。この高速アダマール変換を用いれば  $U(B)$  と  $V'(B)$  を高速に計算することができる。

【0070】次に式 (36) および式 (37) で定義される  $S^{(0)}_m$  と  $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ ) の計算方法について説明する。実数  $x$  をその正負の符号 ( $\pm 1$ ) と、絶対値の対数の組合せで ( $x_0, x_1$ ) ( $x_0 = \text{sgn}(x)$ 、 $x_1 = \log(|x|)$ ) のように表す。また、この表示体系を  $Q = \{(x_0, x_1) \mid x_0 \text{ は } \pm 1, x_1 \text{ は実数}\}$  とおく。この表示体系  $Q$  上では  $Z(B)$  は、下記の式 (44) のように表される。この  $Z(B)$  は高速アダマール変換により算出された  $U(B)$  と  $V'(B)$  から簡単に求められる。

【数39】

$$Z(B_k) = ((-1)^{v(b_k)}, \log V(B_k)) = ((-1)^{v'(B_k)}, V'(B_k)) \quad (44)$$

【0071】ところで、 $Q$  の 2 元  $X = (x_0, x_1)$ 、 $Y = (y_0, y_1)$  の和  $X+Y$  は、この表示形式のもとで次の式 (45) により計算される。ただし、 $f_+$  および  $f_-$

は正の実数上で定義される式 (46) の関数である。  
【数40】

$$\begin{aligned} X+Y &= x_0 \exp(x_1) + y_0 \exp(y_1) = \\ & (x_0, x_1 + f_+(x_1 - y_1)) (x_1 > y_1, x_0 = y_0) \\ & (x_0, x_1 + f_-(x_1 - y_1)) (x_1 < y_1, x_0 = y_0) \\ & (y_0, y_1 + f_+(y_1 - x_1)) (x_1 < y_1, x_0 = y_0) \\ & (y_0, y_1 + f_-(y_1 - x_1)) (x_1 > y_1, x_0 = y_0) \end{aligned} \quad (45)$$

$$\begin{aligned} f_+(x) &= \log(1 + e^{-x}) \\ f_-(x) &= \log(1 - e^{-x}) \end{aligned} \quad (46)$$

【0072】図1は集合  $Q$  の 2 元  $X = (x_0, x_1)$ 、 $Y = (y_0, y_1)$  の和  $X+Y = Z = (z_0, z_1)$  を計算するフローチャートである。まず、ステップ ST11 において加算する  $Q$  の 2 元  $X = (x_0, x_1)$ 、 $Y = (y_0, y_1)$  を設定する。次にステップ ST12 において  $x_1$  が  $y_1$  より大きいかな否かを判定する。 $x_1$  が  $y_1$  より大きい場合はステップ ST13A に進み、 $z_0$ 、 $z_1$ 、 $z_2$  にそ

れぞれ  $x_0$ 、 $x_1$ 、 $x_1 - y_1$  を代入してステップ ST14 に進む。一方、 $x_1$  が  $y_1$  より小さい場合はステップ ST13B に進み、 $z_0$ 、 $z_1$ 、 $z_2$  にそれぞれ  $y_0$ 、 $y_1$ 、 $y_1 - x_1$  を代入してステップ ST14 に進む。

【0073】ステップ ST14 では  $x_0$  と  $y_0$  が等しいかな否かを判定する。 $x_0$  と  $y_0$  が等しい場合はステップ

ST15Aに進み、異なる場合はステップST15Bに進む。ステップST15Aでは変数 $z_1$ に $z_1 + f_+(z_2)$ を代入してステップST16に進む。一方、ステップST15Bでは変数 $z_1$ に $z_1 + f_-(z_2)$ を代入してステップST16に進む。ステップST16ではZに $(z_0, z_1)$ を代入して終了する。

【0074】図1の加算アルゴリズムを用いると、 $S^{(0)}_m$ と $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ )を表示体系Q上で、即ち、対数領域上で計算することができる。図2は $S^{(0)}_m$ と $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ )の計算を示すフローチャートである。ステップST21の $S_0$ と $S_1$ は計算の途中結果を格納するための変数であり、 $k$ はカウンタである。ステップST22の $B_k(m)$ は双対符号 $C'$ の符号語 $B_k = (b_1, b_2, \dots, b_{N-1}, b_N)$  ( $b_m=0, 1$ )の第 $m$ 番目のビット $b_m$ を表す。また、ステップST23およびST24の $z_{k0}, z_{k1}$ は符号語 $B_k$ に対応するZ( $B_k$ )を表示体系Q上で表したときの符号成分と対数成分、即ち、 $Z(B_k) = (z_{k0}, z_{k1})$ である。

【0075】次に図2のフローチャートの動作について説明する。ステップST21において変数 $S_0$ と $S_1$ に

$$P = (p_0, p_1) = (x_0, x_1) + (y_0, y_1)$$

$p_1$ と $q_1$ から次の式(48)により外部値 $L_0(w_m)$ を計算する。また、計算された外部値 $L_0(w_m)$ から次の式(49)により符号ビット $w_m$ のLLR( $w_m$ )を計算する。計算されたLLR( $w_m$ )の符号から符号ビット

$$L_0(w_m) = p_1 - q_1$$

$$LLR(w_m) = L(w_m, y_m) + L_0(w_m)$$

【0078】この実施の形態1の線形符号の軟入力軟出力復号方法は以上のように構成されるので、符号ビットの外部値の計算において双対符号のトレリス線図を用いるのではなく、より簡易なアダマル変換を用いて計算するので、計算量が少なく済み、高速な復号が可能である。また、従来技術では再帰式 $\alpha$ および $\beta$ を記憶するためのメモリが必要であったが、この実施の形態1の復号方法ではZ( $B$ ) ( $B$ は双対符号の符号語)を記憶しておくだけでよい従来技術に比べてメモリ容量を大幅に削減できる効果がある(アダマル行列の対称性から双対符号の符号語半分のZ( $B$ )を記憶すれば十分である。)

【0079】また、 $(N, K)$ 拡大ハミング符号を要素符号とする積符号を復号する場合には、各要素符号の軟入力軟出力復号にこの実施の形態1で説明した復号方法を適用すれば、より高速に積符号の軟入力軟出力繰り返し復号を行うことができる。なお、この実施の形態1では $(N, K)$ 拡大ハミング符号の軟入力軟出力復号方法について説明したが、2元線形ブロック符号の軟入力軟出力復号方法は拡大ハミング符号のみならず、BCH符

Qの要素(1, C)を代入する。ただし、Cは十分大きな負の数である。また、カウンタ $k$ に1を代入する。ステップST22において $B_k(m)$ が0であるか否かを判定する。 $B_k(m)$ が“0”である場合はステップST23に進み、 $B_k(m)$ が“1”である場合はステップST24に進む。ステップST23では変数 $S_0$ と $(z_{k0}, z_{k1})$ の和を変数 $S_0$ に代入し、ステップST26に進む。一方、ステップST24では変数 $S_0$ に $z_{k0} \cdot (-1)^{u_m}$ を、また変数 $S_1$ に $z_{k1} - v'_m$ を代入し、ステップST25に進む。

【0076】ステップST25では変数 $S_1$ と $(x_0, x_1)$ の和を変数 $S_1$ に代入し、ステップST26に進む。ステップST26ではカウンタ $k$ の値が $2N$ に等しいか否かを判定する。カウンタ $k$ の値が $2N$ 未満であればステップST27に進み、カウンタ $k$ をインクリメントしてステップST22以降の処理を繰り返す。一方、カウンタ $k$ の値が $2N$ に等しい場合は処理を終了する。

【0077】計算された $S^{(0)}_m = (x_0, x_1)$ 、 $S^{(1)}_m = (y_0, y_1)$ から下記の式(47)のPおよびQを計算する。

【数41】

$$(47)$$

$w_m$ を式(5)にしたがって推定し、推定情報ビットとして出力する。

【数42】

$$(48)$$

$$(49)$$

号やリード・マラー符号など、他の2元線形ブロック符号に適用できることは言うまでもない。

【0080】実施の形態2. 図3はこの発明の実施の形態2による軟入力軟出力復号装置を示す構成図であり、図において、11A、11Bは整数の加算回路、12は入力される整数を2組の整数に変換して出力する変換テーブル、13Aは第1の高速アダマル変換回路、13Bは第2の高速アダマル変換回路、14は外部値計算回路、15は情報ビットを推定する判定回路である。

【0081】次に図3の復号装置の動作について、上記実施の形態1と同じ $(N, K)$ 拡大ハミング符号を用いて説明する。まず、復調器から入力される通信路値 $y_m$ と事前値 $L_{a_m}$ が加算回路11Aにおいて加算されて式(9)の軟入力値 $L(w_m, y_m)$  ( $m=1, 2, \dots, N-1, N$ )が生成される。ただし、入力される通信路値 $y_m$ と事前値 $L_{a_m}$ は量子化されて整数として表されているものとする。

【0082】生成された軟入力値 $L(w_m, y_m)$ は変換テーブル12に入力され、式(27)に示す $u_m$ と式(28)に示す $v_m$ の対数 $v'_m = \log v_m$ が生成さ

れる。軟入力値  $L(w_m, y_m)$  が正であれば  $u_m$  に 0、負であれば  $u_m$  に 1 がセットされ、また、 $x = L(w_m, y_m)$  から  $x$  の関数  $f(x) = \log(\tanh(|x|))$  の量子化値を格納したテーブルを索表して  $v'_m$  に  $\log(\tanh(|L(w_m, y_m)|))$  がセットされて出力される。系列  $v'_m$  は第 1 の高速アダマール変換回路 13 A および外部値計算回路 14 に入力され、系列  $u_m$  は第 2 の高速アダマール変換回路 13 B および外部値計算回路 14 に入力される。

【0083】高速アダマール変換回路 13 A、13 B はグリーン・マシーンやその他の公知の技術で構成される。高速アダマール変換回路 13 A では式 (42) の  $V'_k = V'(B_k)$  ( $k=1, 2, \dots, 2N$ ) が生成され、高速アダマール変換回路 13 B では式 (30) の  $U_k = U(B_k)$  ( $k=1, 2, \dots, 2N$ ) が生成される。高速アダマール変換回路 13 A、13 B において生成された系列  $V'_k$  および系列  $U_k$  は外部値計算回路 14 に入力される。

【0084】外部値計算回路 14 では変換テーブル 12 から出力される軟入力値の変換信号  $v'_m$  と  $u_m$ 、高速アダマール変換回路 13 A から出力される系列  $V'_k$ 、高速アダマール変換回路 13 B から出力される系列  $U_k$  がそれぞれ入力されて式 (35) の外部値が計算される。まず、 $U_k$  が偶数ならば  $z_{k_0}$  に 1、奇数ならば  $z_{k_0}$  に -1 がセットされ、 $z_{k_1}$  に  $V'_k$  がセットされて式 (44) の  $Z(B_k)$  の Q 表現  $Z_k = (z_{k_0}, z_{k_1})$  が生成される。

【0085】生成された  $Z_k = (z_{k_0}, z_{k_1})$  および軟入力値の変換信号  $v'_m$  および  $u_m$  から式 (36) および式 (37) の  $S^{(0)}_m$  と  $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ ) を計算する回路を図 4 に示す。図中の  $r_{0_0}$  と  $r_{0_1}$  は  $S^{(0)}_m$  の Q 表現  $S^{(0)}_m = (r_{0_0}, r_{0_1})$  の各成分を格納するレジスタ、 $r_{1_0}$  と  $r_{1_1}$  は  $S^{(1)}_m$  の Q 表現  $S^{(1)}_m = (r_{1_0}, r_{1_1})$  の各成分を格納するレジスタであり、 $B_k(m)$  は双対符号  $C'$  の符号語  $B_k = (b_1, b_2, \dots, b_{N-1}, b_N)$  ( $b_m=0, 1$ ) の第  $m$  番目のビット  $b_m$  である。

【0086】また、31 は Q 表現の Q 加算回路、32 A は Q 加算回路 31 の出力をレジスタ  $r_{0_1}$  または  $r_{1_1}$  に入力するスイッチ、32 B はレジスタ  $r_{0_1}$  または  $r_{1_1}$  の出力を選択して Q 加算回路 31 に入力するスイッチ、32 C は Q 加算回路 31 の出力をレジスタ  $r_{0_0}$  または  $r_{1_0}$  に入力するスイッチ、32 D はレジスタ  $r_{0_0}$  または  $r_{1_0}$  の出力を選択して Q 加算回路 31 に入力するスイッチ、33 A、33 B はセレクタ、34 は整数の加算回路、35 は論理積ゲート、36 A、36 B は -1 倍回路である。

【0087】図 4 の回路の動作を説明する前に Q 加算回路 31 の構成および動作について説明する。図 5 は Q の 2 元  $X = (x_0, x_1)$  と  $Y = (y_0, y_1)$  を加算する Q

加算回路の構成を示す回路図であり、図において、41 A、41 B は整数の加算回路、42 A、42 B、42 C、42 D はセレクタ、43 A は式 (46) の関数  $f_+(x)$  の量子化値を格納したルックアップテーブル、43 B は式 (46) の関数  $f_-(x)$  の量子化値を格納したルックアップテーブル、44 は入力される値 ( $w$ ) が正であるか否かを判定する判定回路、45 は入力される  $x_0$  と  $y_0$  が等しいか否かを判定する判定回路、46 A、46 B は -1 倍回路である。

【0088】次に図 5 の動作について説明する。 $x_1$  は加算回路 41 A およびセレクタ 42 B の入力 A に入力され、 $x_0$  はセレクタ 42 C の入力 A および判定回路 45 に入力される。一方、 $y_1$  は -1 倍回路 46 A およびセレクタ 42 B の入力 B に入力され、 $y_0$  はセレクタ 42 C の入力 B および判定回路 45 に入力される。判定回路 45 では  $x_0$  と  $y_0$  が等しい場合 1 が出力され、等しくない場合は 0 が出力される。

【0089】また、-1 倍回路 46 A では  $y_1$  が -1 倍されて、 $-y_1$  が加算回路 41 A に出力される。加算回路 41 A では  $x_1$  と  $-y_1$  が加算されて、 $x_1 - y_1$  が判定回路 44、セレクタ 42 A の入力 A、-1 倍回路 46 B に入力される。-1 倍回路 46 B では  $x_1 - y_1$  が -1 倍されて、 $-x_1 + y_1$  がセレクタ 42 A の入力 B に入力される。判定回路 44 では  $w = x_1 - y_1$  が正であれば 1、負であれば 0 がセレクタ 42 A、42 B、42 C に出力される。セレクタ 42 A では判定回路 44 から入力される信号が 1 であれば入力 A が選択され、0 であれば入力 B が選択されてルックアップテーブル 43 A、43 B に入力される。

【0090】ルックアップテーブル 43 A では、入力される  $x$  からテーブルを索表して対応する  $f_+(x)$  がセレクタ 42 D の入力 A に出力され、ルックアップテーブル 43 B では、入力される  $x$  からテーブルを索表して対応する  $f_-(x)$  がセレクタ 42 D の入力 B に出力される。セレクタ 42 B では、判定回路 44 から入力される信号が 1 であれば入力 A が選択され、0 であれば入力 B が選択されて加算回路 41 B に出力される。また、セレクタ 42 D では、判定回路 45 の出力信号が 1 であれば入力 A が選択され、0 であれば入力 B が選択されて加算回路 41 B に出力される。加算回路 41 B では、セレクタ 42 B の出力とセレクタ 42 D の出力が加算されて  $z_1$  が生成される。一方、セレクタ 42 C では、判定回路 44 の出力信号が 1 であれば入力 A が選択され、0 であれば入力 B が選択されて  $z_0$  が生成される。

【0091】次に図 4 の回路の動作について説明する。各  $m$  ( $m=1, 2, \dots, N-1, N$ ) ごとに図 4 の回路が並列に配置されて動作するが、ここでは  $m$  を 1 つ固定して説明する。軟入力値の変換信号の 1 つ  $v'_m$  は -1 倍回路 36 A で -1 倍されて  $-v'_m$  が生成され、加算回路 34 に入力される。また、もう 1 つの変換信号  $u_m$

は論理積ゲート35に入力される。 $Z_k = (z_{k0}, z_{k1})$ が $k=1$ から $2N$ まで順に入力され、 $z_{k1}$ はセクタ33Aの入力Aおよび加算回路34に入力され、 $z_{k0}$ はセクタ33Bの入力Aおよび-1倍回路36Bに入力される。また、 $B_k(m)$ はセクタ33Aに入力される。

【0092】加算回路34では、 $z_{k1}$ と $-v'_m$ が加算されて $z_{k1} - v'_m$ がセクタ33Aの入力Bに入力される。セクタ33Aでは、 $B_k(m)$ が0のとき入力Aが選択され、1のとき入力Bが選択されてQ加算回路31に入力される（入力 $x_1$ ）。-1倍回路36Bでは、 $z_{k0}$ が-1倍されて $-z_{k0}$ がセクタ33Bに入力される。セクタ33Bでは、 $B_k(m)$ と $u_m$ の論理積が0のときは入力Aが選択され、1のときは入力Bが選択されてQ加算回路31に入力される（入力 $x_0$ ）。

【0093】スイッチ32Bでは、 $B_k(m)$ が0のとき端子0に接続されてレジスタ $r_{01}$ の出力が、一方、 $B_k(m)$ が1のとき端子1に接続されてレジスタ $r_{11}$ の出力がQ加算回路31に入力される（入力 $y_1$ ）。また、スイッチ32Dでは、 $B_k(m)$ が0のとき端子0に接続されてレジスタ $r_{00}$ の出力が、一方、 $B_k(m)$ が1のとき端子1に接続されてレジスタ $r_{10}$ の出力がQ加算回路31に入力される（入力 $y_0$ ）。

【0094】Q加算回路31の出力 $z_1$ は、スイッチ32Aを介してレジスタ $r_{01}$ または $r_{11}$ に入力される。 $B_k(m)$ が0のときスイッチ32Aは端子0に接続されてQ加算回路31の出力 $z_1$ はレジスタ $r_{01}$ に入力され、 $B_k(m)$ が1のときスイッチ32Aは端子1に接続されてQ加算回路31の出力 $z_1$ はレジスタ $r_{11}$ に入力される。

【0095】また、Q加算回路31の出力 $z_0$ は、スイッチ32Cを介してレジスタ $r_{00}$ または $r_{10}$ に入力される。 $B_k(m)$ が0のときスイッチ32Cは端子0に接続されてQ加算回路の出力 $z_0$ はレジスタ $r_{00}$ に入力され、 $B_k(m)$ が1のときスイッチ32Cは端子1に接続されてQ加算回路31の出力 $z_0$ はレジスタ $r_{10}$ に入力される。

【0096】 $Z_{2N} = (z_{2N0}, z_{2N1})$ の入力が入った段階でレジスタの組 $(r_{00}, r_{01})$ に格納されている内容が $S^{(0)}_m$ であり、レジスタの組 $(r_{10}, r_{11})$ に格納されている内容が $S^{(1)}_m$ である。 $S^{(0)}_m = (r_{00}, r_{01})$ および $S^{(1)}_m = (r_{10}, r_{11})$ から、Qの加算 $(p_0, p_1) = (r_{00}, r_{01}) + (r_{10}, r_{11})$ および $(q_0, q_1) = (r_{00}, r_{01}) + (-r_{10}, r_{11})$ が計算されて符号ビット $w_m$ の外部値 $Le(w_m) = p_1 - q_1$ が算出される。

【0097】図3に示す外部値計算回路14で計算された符号ビット $w_m$ の外部値 $Le(w_m)$ は軟出力として出

力されるとともに、加算回路11Bに出力される。加算回路11Bでは、加算回路11Aの軟入力値 $L(w_m, y_m)$ と外部値計算回路14で計算された外部値 $Le(w_m)$ が加算されて $LLR_m$ が生成される。生成された $LLR_m$ は判定回路15に入力されて正であるか否かが判定される。正であれば $w_m$ に0がセットされ、負であれば $w_m$ に1がセットされて出力される。

【0098】この実施の形態2の2元線形ブロック符号の軟入力軟出力復号装置は、上述したように高速アダマール変換を利用して高速に外部値を計算できる効果がある。また、積符号を軟入力軟出力繰り返し復号する場合に、要素符号の復号に軟入力軟出力復号装置を適用すれば高速な積符号の復号装置が構成できる。

【0099】

【発明の効果】以上のように、この発明によれば、生成ステップで生成された硬判定データをアダマール変換して第1のデータ系列を生成し、その信頼度情報をアダマール変換して第2のデータ系列を生成するアダマール変換ステップを設けるように構成したので、2元線形ブロック符号の軟入力軟出力復号を少ない計算量とメモリ容量で行うことができる効果がある。

【0100】この発明によれば、事後値計算ステップが2元線形ブロック符号C1、C2から構成された積符号を復号する場合、C1符号の事後値を計算すると、その事後値をC2符号の事前値として軟入力値計算ステップに与え、C2符号の事後値を計算すると、その事後値をC1符号の事前値として軟入力値計算ステップに与えるように構成したので、積符号の軟入力軟出力復号を少ない計算量とメモリ容量で行うことができる効果がある。

【0101】この発明によれば、軟入力値変換回路により生成された硬判定データをアダマール変換して第1のデータ系列を生成し、その信頼度情報をアダマール変換して第2のデータ系列を生成するアダマール変換回路を設けるように構成したので、2元線形ブロック符号の軟入力軟出力復号を少ない計算量とメモリ容量で行うことができる効果がある。

【0102】この発明によれば、事後値計算回路が2元線形ブロック符号C1、C2から構成された積符号を復号する場合、C1符号の事後値を計算すると、その事後値をC2符号の事前値として軟入力値計算回路に与え、C2符号の事後値を計算すると、その事後値をC1符号の事前値として軟入力値計算回路に与えるように構成したので、積符号の軟入力軟出力復号を少ない計算量とメモリ容量で行うことができる効果がある。

【図面の簡単な説明】

【図1】 集合Qの2元 $X = (x_0, x_1)$ 、 $Y = (y_0, y_1)$ の和 $X + Y = Z = (z_0, z_1)$ を計算するフローチャートである。

【図2】  $S^{(0)}_m$ と $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ )の計算を示すフローチャートである。



【図3】 この発明の実施の形態2による軟入力軟出力復号装置を示す構成図である。

【図4】  $S^{(0)}_m$  と  $S^{(1)}_m$  ( $m=1, 2, \dots, N-1, N$ ) を計算する回路を示す構成図である。

【図5】 Q加算回路の構成を示す回路図である。

【図6】 積符号の一般的な構成を示す説明図である。

【図7】 一般的なディジタル通信システムを示す構成図である。

【図8】 (7, 3) 巡回符号の符号器の構成を示すブロック図である。

【図9】 (7, 3) 巡回符号のトレリス線図を示す説明図である。

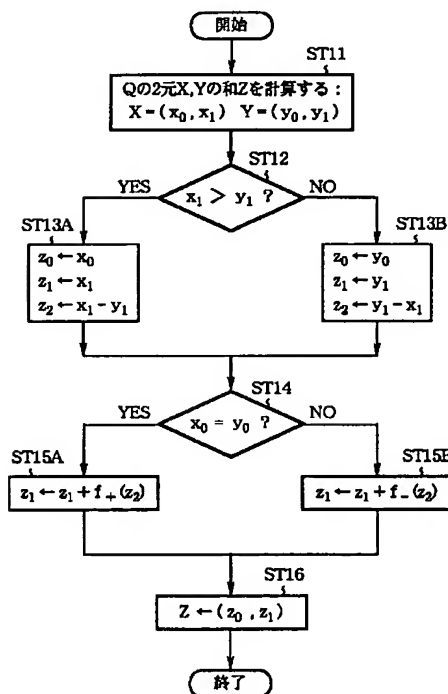
【図10】 積符号の軟入力軟出力繰返し復号法を示

すフローチャートである。

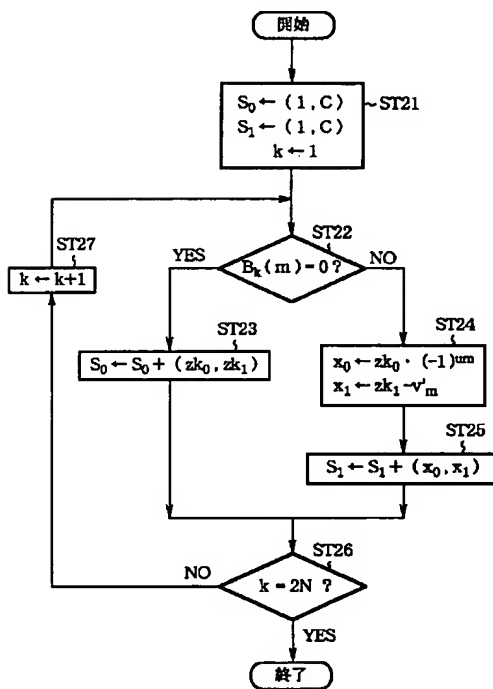
【符号の説明】

11A, 11B 整数の加算回路、12 変換テーブル、13A 第1の高速アダマール変換回路、13B 第2の高速アダマール変換回路、14 外部値計算回路、15 判定回路、31 Q加算回路、32A, 32B, 32C, 32D スイッチ、33A, 33B セレクタ、34 整数の加算回路、35 論理積ゲート、36A, 36B -1倍回路、41A, 41B 整数の加算回路、42A, 42B, 42C, 42D セレクタ、43A, 43B ルックアップテーブル、44 判定回路、45 判定回路、46A, 46B -1倍回路。

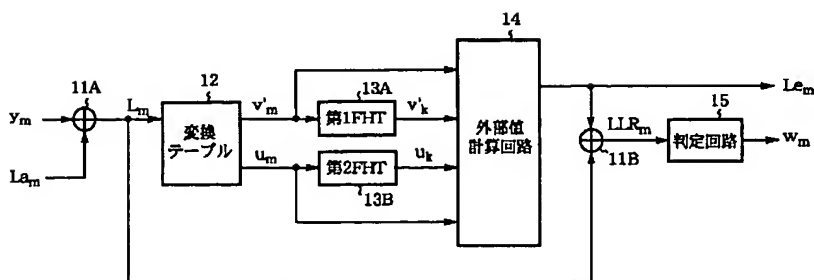
【図1】



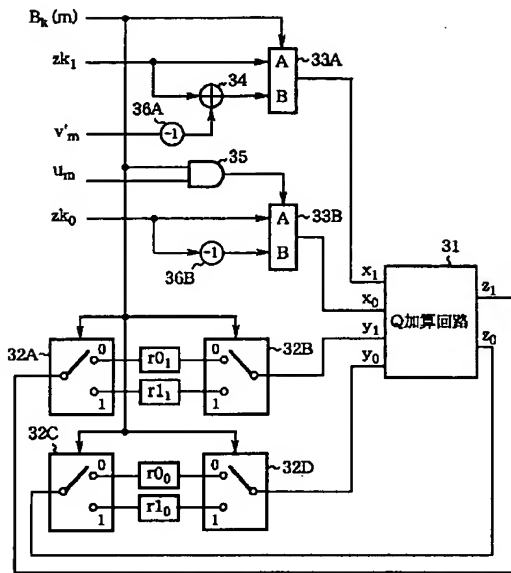
【図2】



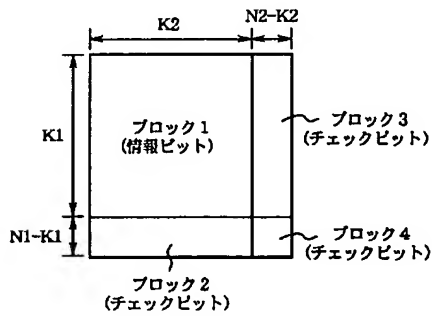
【図3】



【図4】

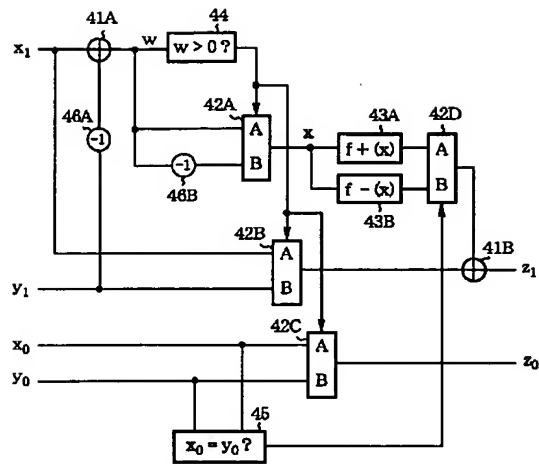


【図6】

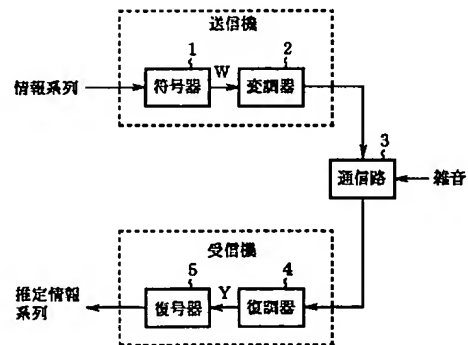


【図8】

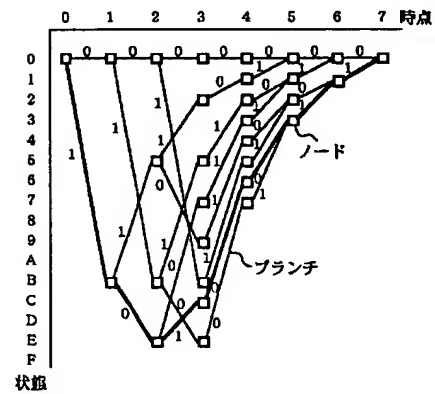
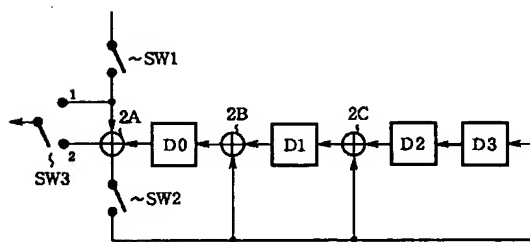
【図5】



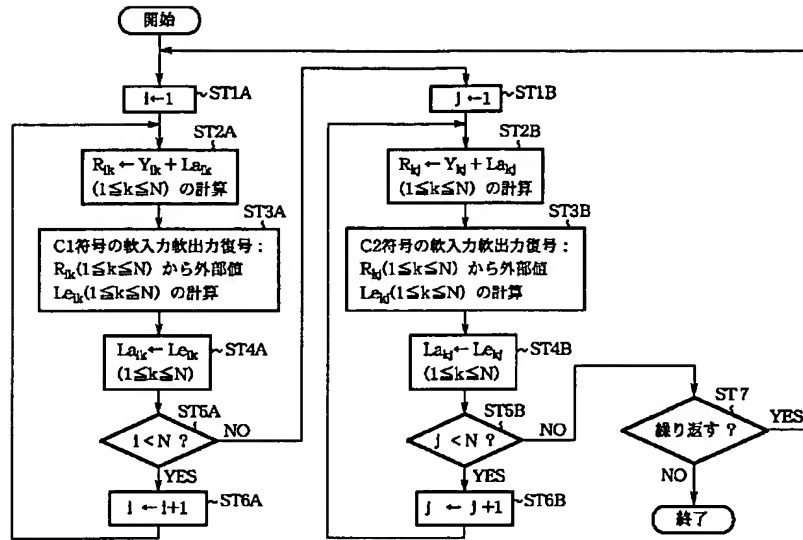
【図7】



【図9】



【図10】



フロントページの続き

(72)発明者 中村 隆彦

東京都千代田区丸の内二丁目2番3号 三  
菱電機株式会社内

(72)発明者 吉田 英夫

東京都千代田区丸の内二丁目2番3号 三  
菱電機株式会社内

Fターム(参考) 5B001 AA01 AA08 AA13 AB02 AB03  
AC01 AD06  
5J065 AC02 AD05 AF03 AG05 AH02  
AH07 AH21